Express Mail EU943994330US

APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: Todd E. Leonard *et al.*

TITLE: SCALABLE CYCLIC REDUNDANCY CHECK CIRCUIT

DOCKET NO.: BUR920030101US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

SCALABLE CYCLIC REDUNDANCY CHECK CIRCUIT

FIELD OF THE INVENTION

The present invention relates to the field of cyclic redundancy check circuits; more specifically, it relates to a scalable cyclic redundancy check circuit.

BACKGROUND OF THE INVENTION

Error checking of data transmissions between sending and receiving devices use a cyclic redundancy check circuit (CRC) implementing various CRC codes in both the sending and receiving devices. The CRC code is calculated by an exclusive OR (XOR) subtree. As high speed serial interconnect technologies evolve, many of the standards governing these technologies allow bandwidths well beyond the traditional 96 and 128 bits per cycle bandwidths, yet maintain the same transmission frequency as for the older smaller 96 and 128 bits per cycle bandwidths. As bandwidth increases, the complexity and depth of the XOR subtree must increase as the need to process more bits per clock cycle grows. Tradition CRC designs when applied to large bandwidth data transmissions very quickly develop the interrelated problems of timing closure and physical silicon area required to implement the XOR subtree. Therefore, there is a need for a CRC circuit that can handle large bandwidths without timing closure problems.

SUMMARY OF THE INVENTION

A first aspect of the present invention is a cyclic redundancy check circuit, comprising: a W-bit packet data slice latch having outputs; a multiple level XOR subtree having inputs and outputs, each level comprising one or more XOR subtrees, each output of the packet data slice latch coupled to an input of the multiple level XOR subtree, each lower level XOR subtree of the multiple level XOR subtree coupled to a higher level XOR subtree of the multiple level XOR subtree through an intervening latch level; a remainder XOR subtree having inputs and outputs; a combinational XOR subtree having inputs and outputs, the outputs of the remainder XOR subtree and the outputs of the multiple level XOR subtree coupled to the inputs of the combinational XOR subtree; and an M-bit current CRC result latch having inputs and outputs, the output of the combinational XOR subtree coupled to the inputs of the current CRC result latch and to the inputs of the remainder XOR subtree.

A second aspect of the present invention is a method for cyclic redundancy check calculation, comprising: providing a W-bit packet data slice latch having outputs; providing a multiple level XOR subtree having inputs and outputs, each level comprising one or more XOR subtrees, each output of the packet data slice latch coupled to an input of the multiple level XOR subtree, each lower level XOR subtree of the multiple level XOR subtree coupled to a higher level XOR subtree of the multiple level XOR subtree through an intervening latch level; providing a remainder XOR subtree having inputs and outputs; providing a combinational XOR subtree having inputs and outputs, the outputs

of the remainder XOR subtree and the outputs of the multiple level XOR subtree coupled to the inputs of the combinational XOR subtree; and providing an M-bit current CRC result latch having inputs and outputs, the output of the combinational XOR subtree coupled to the inputs of the current CRC result latch and to the inputs of the remainder XOR subtree.

A third aspect of the present invention is a method of designing an M-bit cyclic redundancy check circuit, the method comprising: partitioning an XOR function of the cyclic redundancy check circuit into a remainder XOR partition and a multiple level packet data slice XOR partition; determining I, the largest number of bits I of a subset of the M-bits of a CRC result required to generate output bits of a remainder partition XOR subtree of the cyclic redundancy check circuit; determining Z, the largest number of inputs to an XOR gate in a design library to be used in the cyclic redundancy check circuit; calculating K, the maximum number of XOR stages comprised of Z-input XOR gates in the remainder XOR subtree; calculating N, the maximum number of inputs to any XOR subtree in any level of a multiple level XOR subtree partition of the cyclic redundancy check circuit; partitioning the multiple level XOR subtree partition into XOR subtrees having no number of inputs that is larger than a number of inputs to the remainder XOR subtree; and inserting a latch between each XOR subtree of a lower level partition of the packet data slice XOR partition and a immediately higher level partition of the packet data slice XOR partition.

BRIEF DESCRIPTION OF DRAWINGS

The features of the invention are set forth in the appended claims. The invention itself, however, will be best understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

FIG. 1 is an exemplary related art 32-bit CRC circuit.

FIG. 2 is an exemplary 32-bit CRC circuit according to the present invention;

FIG. 3 is a generic scalable M-bit CRC circuit according to the present invention; and

FIG. 4 is a flowchart of the method of designing a scalable CRC circuit according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The terminology Q by P-way XOR subtree defines an XOR subtree having Q outputs and (P x Q) inputs. The notation Q^P should be read as Q^P.

FIG. 1 is an exemplary related art 32-bit CRC circuit. In FIG. 1, a CRC circuit **100** includes a 128-bit packet data slice latch **105**, a single, 160-bit input/32-bit output XOR tree **110** and a 32-bit current CRC remainder latch **115**. The bit width of the remainder latch defines the CRC type, in the present example a 32-bit CRC or CRC32. The output of packet slice latch **105** is connected to the input of XOR tree **110** by a 128-bit bus; each bit being connected to a different input. The output of XOR subtree **110** is connected to the input of current CRC remainder latch **115**. The output of current CRC

remainder latch **115** is a 32-bit CRC output bus **120**, which is also connected to the input of XOR tree **110** to provide the cyclic portion of the CRC result. Each output bit of current CRC remainder latch **115** is connected to a different input of XOR tree **110**, and to inputs not used by packet data slice latch **105**.

Data bits are moved from packet data slice latch **105** through XOR tree **110** and current CRC remainder latch **115** by a clock signal CLK. The same CLK signal moves data bits out of current CRC remainder latch **115** onto CRC output bus and into XOR tree **110**. The arrangement of XOR gates in XOR tree **110** implements the CRC code and performs the actual CRC calculation.

As the number of input bits to an XOR tree increases, the depth of XOR gates (the number of XOR gates connected in series from the input to the output of the XOR tree) as well as the number of inputs in each individual XOR gate in the XOR tree increases. At some point, it will take more than a single clock cycle for data bits to travel through the XOR tree and the CRC circuit will generate an erroneous CRC result. The present invention avoids XOR tree data bit propagation time problems by partitioning the XOR tree into XOR-subtrees, which are each small enough not to have a data bit propagation time problem. It should be noted that data bit propagation time is dependent on the integrated circuit technology that the CRC circuit is physically fabricated in.

The present invention partitions the XOR tree into two main partitions. The first partition is a single XOR subtree for processing the remainder of the CRC. The second partition is a multi-level partition, each level comprised of multiple XOR subtrees. Each

of these multiple XOR subtrees is no larger than the remainder XOR subtree. Each level of XOR subtrees perform a portion of the CRC calculation and each XOR subtree belonging to a particular level performs a portion of the portion of the CRC calculation performed by the level. The size of the remainder subtree is chosen so that all the XOR calculation it performs can be completed in one clock cycle. Since all the XOR subtrees of the multi-level partition are the size (or smaller) each levels portion of the CRC is likewise performed in one clock cycle or less.

FIG. 2 is an exemplary CRC circuit according to the present invention. In FIG. 2, a CRC circuit **200** includes a 2048-bit packet data slice latch **205**, 27 sets (in subsets of 8) of 32 by (0 to 5)-way leaf XOR subtrees **210** and corresponding 32-bit latches **215**, 27 sets of 32 by 8-way XOR subtrees **220** and corresponding 32-bit latches **225**, a 32 by 27-way XOR subtree **230**, a 32 by 2-way XOR subtree **235**, a remainder XOR subtree **240** and a 32-bit current CRC remainder latch **245**. Packet data slice latch **205** is a partition level 0 latch. Latches **215** are partition level 1 latches, and latches **225** are partition level 2 latches, so there are three latch levels in CRC circuit **200**. Leaf XOR subtrees **210**, XOR subtrees **220** and **230** may be considered to be in a data slice XOR subtree.

Each leaf XOR subtree **210** is connected to packet data slice latch **205** by 0 to 5 32-bit inputs (i. e. 160 inputs to each leaf XOR subtree). Each of the 32 outputs of each leaf XOR subtree **210** is connected to a different input of a corresponding latch **215**. There need not be any particular relationship between a particular input of a particular leaf XOR subtree **210** and a particular bit from packet data slice latch **205**. Each of the 32

outputs of each latch **215** of each set of 8 latches **215** is connected to a different input of a corresponding XOR subtree **220**. Each of the 32 outputs of each latch **225** is connected to a different input of XOR subtree **230**. Each of the 32 outputs of XOR subtree **230** is connected to a different input of a 32 member first subset of the 64 inputs of XOR subtree **235**. Each of the 32 outputs of XOR subtree **235** is connected to a different input of current CRC remainder latch **245**. The 32 outputs of current CRC remainder latch **245** are connected to a 32-bit output bus **250** and to a different input of remainder XOR subtree **240**. Each of the 32 outputs of remainder XOR subtree **245** is connected to a different input of a second 32 member of the 64 inputs of XOR subtree **235**. The two subsets do not have common inputs.

Data bits are moved from packet data slice latch **205** through leaf XOR subtrees **210** into latches **215** by clock signal CLK. Data bits are moved from latches **215** through XOR subtrees **220** and into latches **225** by clock signal CLK. Data bits are moved from latches **225**, through XOR subtrees **230** and **235** into current CRC remainder latch **245** by clock signal CLK Data bits are moved from current remainder latch **245** onto output bus **250** and through remainder XOR subtree **240** and XOR subtree **235** back into current CRC remainder latch **245** by clock signal CLK. The specific arrangement of XOR gates in leaf XOR subtree **210** and XOR subtrees **220, 230, 235** and **240** implements the CRC code and performs the actual CRC calculation.

The structure of CRC circuit **200** is determined by maximum delay through the XOR subtree **240**. For example, if XOR subtree **240** is implemented using only 3-input

and 2-input XOR gates and the largest CRC remainder expected is 1059-bits then the maximum size of a subset of the 32-bit CRC remainder is 20-bits. The value 1059 is specific to the particular CRC calculation and number of bits processed per CLK cycle. The value 20 is also determined by the particular CRC calculation as are the particular the bits of the 32-bit input to remainder XOR subtree **240** in the subset. The XOR gate structure containing the shortest delay path is realized in a 3 (the smallest whole positive number greater than $\log_3 20$) XOR gate level XOR subtree. The maximum number of inputs of a 3 XOR gate level XOR subtree using 3 input XOR gates is 3^3 or 27. Thus when partitioning the XOR subtree comprised of leaf XOR subtrees **210** and XOR subtrees **220**, **230** and **235**, each partition must not be larger than a 27 input XOR operation. The minimum number of latch stages in the XOR subtree comprised of leaf XOR subtrees **210** and XOR subtrees **220**, **230** and **235** is 3 (the smallest whole positive number greater than $\log_{27} 1059$). To process 2048-bits of data in one clock cycle, the worst-case single XOR operation must operate on 1059 bits.

A data packet's 32-bit CRC remainder is calculated by initializing CRC **200** to a value of 0xFFFF_FFFF, and then processing the packet through the CRC circuit. Given the current CRC remainder value and a 2048-bit slice of the data packet, the next CRC remainder is calculated and then latched. The next CRC remainder value is calculated by performing a bit wise XOR operation on the two 32-bit outputs of XOR subtree **235** and remainder XOR subtree **240**. Each bit of the output of remainder XOR subtree **240** is calculated by performing an XOR operation over a subset of bits of the current CRC

remainder value. Each bit of the output of XOR subtree **230** is calculated by performing an XOR operation over a subset of bits of the portion of packet data currently being processed.

While the output of remainder XOR subtree **240** is the result a single XOR operation, the output XOR subtree **230** is the result of several levels or partitions of XOR operations performed respectively by XOR subtree **230,** XOR subtrees **220** and leaf XOR subtrees **210**. The topmost XOR operation partition (that performed by XOR subtree **230**) is picked such that each output is fed by an XOR operation on 27 inputs. The remaining, lower XOR operation partition sizes (those performed by XOR subtrees **220** and by leaf XOR subtrees **210**) are picked arbitrarily to balance partition sizes across the bottom two partitions. There are 244 partitions total. (8X27=216 level 0 partitions, 27 level 1partitions and 1 level 2 partition.) The output of each partition, except for the last partition, is latched. When the last 2048-bits of a data packet are processed, the next CRC remainder is the CRC value for the packet.

FIG. 3 is a generic scalable M-bit CRC circuit according to the present invention. In FIG. 3, a CRC circuit **300** includes a K-bit packet data slice latch **305**, N^Y M by N-way leaf XOR subtrees **310** and corresponding M-bit latches **315** (N, Y and M are defined *infra*), intermediate levels of M by N-way XOR subtrees and corresponding latches (not shown), N^2 of M by N-way XOR subtrees **320** and corresponding M-bit latches **325**, N M by N-way XOR subtrees **330** and corresponding M-bit latches **335**, a M by N way XOR subtree **340**, an M by 2 way XOR subtree **345**, a remainder XOR subtree **350** and an M-

bit current CRC remainder latch **360**. Packet data slice latch **305** is a partition level 0 latch. Latches **315** are partition level 1 latches, latches **315** are partition level (Y-1) latches and latches **335** are partition level Y latches, so there are Y+1 partition levels in CRC circuit **300**. Leaf XOR subtrees **310**, intermediate XOR subtrees (not shown), XOR subtrees **320**, **325** and **340** may be considered to be in a data slice XOR subtree.

Each leaf XOR subtree **310** is connected to packet data slice latch **305** by variable numbers of M-bit input. Each of the M outputs of each leaf XOR subtree **310** is connected to a different input of a corresponding latch **315**. There need not be any particular relationship between a particular input of a particular leaf XOR subtree **310** and a particular bit from packet data slice latch **305**. After progressing through intermediate partition levels, each of the M outputs of each of XOR subtrees **320** is connected to a different input of corresponding latches **325**. Each of the M outputs of each latch **325** is connected to a different input of corresponding XOR subtrees **330**. Each of the M outputs of XOR subtrees **330** are connected a different input of corresponding latches **335**. Each of the M inputs of latches **335** is connected to different inputs of XOR subtree **340**. Each of the M outputs of XOR subtree **340** is connected to a different input of a first M member subset of the 2M inputs of XOR subtree **345**. Each of the M outputs of XOR subtree **345** is connected to a different input of current CRC remainder latch **355**. The M outputs of current CRC remainder latch **355** are connected to an M-bit output bus **360** and to different inputs of remainder XOR subtree **350**. Each of the M outputs of remainder

XOR subtree **350** is connected to a different input of a second M member subset of the 2M inputs of XOR subtree **345**. The two subsets do not have common inputs.

Data bits are moved from packet data slice latch **305** through the (Y-1) partition levels by a clock signal CLK applied to the latches within each partition level. The specific arrangement of XOR gates in the XOR subtrees of the various partition levels of CRC circuit **300** and XOR subtrees **340** and **345** and remainder XOR subtree **350** implements the CRC code and performs the actual CRC calculation.

The structure of CRC circuit **300** is determined by maximum delay through the XOR subtree **350**. For example, if XOR subtree **350** is implemented using only Z-input or smaller XOR gates and the largest CRC remainder expected is J-bits then the maximum size of a subset of the M-bit CRC remainder is I-bits. The value I is specific to the particular CRC calculation and number of bits processed per CLK cycle. The value J is determined by the particular CRC calculation as are the particular bits of the M-bit input to remainder XOR subtree **350** in the subset. The XOR gate structure containing the shortest delay path is realized in a K (the smallest whole positive number greater than $\log_Z I$) XOR gate level XOR subtree. The maximum number of inputs of a K level XOR subtree using Z input XOR gates is $K^Z = N$. Thus when partitioning the XOR subtree comprised of leaf XOR subtrees **310** through XOR subtree **345**, each partition must not be larger than a J input XOR operation (which is the size of XOR operation performed by remainder XOR subtree **340**). The minimum number of latch stages in the XOR subtree

comprised of leaf XOR subtrees **310** through XOR subtree **345** is Y+1 (the smallest whole positive number greater than $\log_N J$).

A data packet's M-bit CRC remainder is calculated by initializing CRC **300** to a value of -1, and then processing the packet through the CRC circuit. Given the current CRC remainder value and a W-bit slice of the data packet, the next CRC remainder is calculated and then latched. The next CRC remainder value is calculated by performing a bit wise XOR operation on the two M-bit outputs of XOR subtree **345** and remainder XOR subtree **350**. Each bit of the output of remainder XOR subtree **350** is calculated by performing an XOR operation over a subset of bits of the current CRC remainder value. Each bit of the output of XOR subtree **340** is calculated by performing an XOR operation over a subset of bits of the portion of packet data currently being processed.

Another way to understand the structure of CRC circuit **300** is that the number of levels of said XOR subtrees in the XOR subtree from leaf XOR subtrees **310** to XOR subtree **340** is a function of A, the maximum number of input bits to all leaf XOR subtrees **310** to give a single output bit of XOR subtree **340** and of B, the maximum number of input bits to remainder XOR subtree **350** to give a single output bit of the remainder XOR subtree. The number of levels of said XOR subtrees in the XOR subtree from leaf XOR subtrees **310** to XOR subtree **340** being $\log_B A$.

While the output of remainder XOR subtree **350** is the result of a single XOR operation, the output XOR subtree **340** is the result of several levels or partitions of XOR operations performed as illustrated in FIG. 3. The topmost XOR operation partition (that

performed by XOR subtree **340**) is picked such that each output is fed by an XOR operation on N inputs. The output of each partition, except for the last partition, is latched. When the last W-bits of a data packet are processed, the next CRC remainder is the CRC value for the packet.

FIG. 4 is a flowchart of the method of designing a scalable CRC circuit according to the present invention. In step **400** it the largest number of bits in a subset of bits of a CRC remainder to be processed by an XOR operation on that subset of bits by the remainder XOR subtree is determined. This number designated I, and the particular bits of the CRC remainder making up the subset is a function of the CRC function being implemented.

In step **405**, from a design library **410** of circuit elements, the XOR gate having the largest number of inputs is determined. Generally this XOR gate is determined by the length of the data path through the XOR gate, its attendant delay, and the amount of integrated circuit real estate it requires. This number of inputs is designated Z.

In step **415**, the largest number of XOR gate levels K, in the remainder XOR subtree is calculated using the formula K = smallest whole positive number greater than $(\log_Z I)$.

In step **420**, the largest number of XOR operations N, that are no slower than the XOR operations performed by the remainder XOR subtree is calculated using the formula $N = K^Z$.

In step **425**, the data slice XOR subtree operating on data packet slices is partitioned into XOR subtrees such that no XOR subtree of the data slice XOR subtree has more inputs then the remainder XOR subtree.

In step **430**, the XOR output of every XOR subtree in the data slice XOR subtree are latched except the topmost XOR subtree.

Thus, the present invention provides a scalable CRC circuit that can handle large bandwidths without timing closure problems.

The description of the embodiments of the present invention is given above for the understanding of the present invention. It will be understood that the invention is not limited to the particular embodiments described herein, but is capable of various modifications, rearrangements and substitutions as will now become apparent to those skilled in the art without departing from the scope of the invention. Therefore it is intended that the following claims cover all such modifications and changes as fall within the true spirit and scope of the invention.